JUN 1991

BEGINNERS' WORKSHOP

This month it's back to basics at the Beginners' Workshop *as Mike Hardaker explains what's involved in building a system disk.*

This month we're going to take a good look at one of the most basic PC tasks that you are likely to come across – making an MS-DOS 'system' disk.

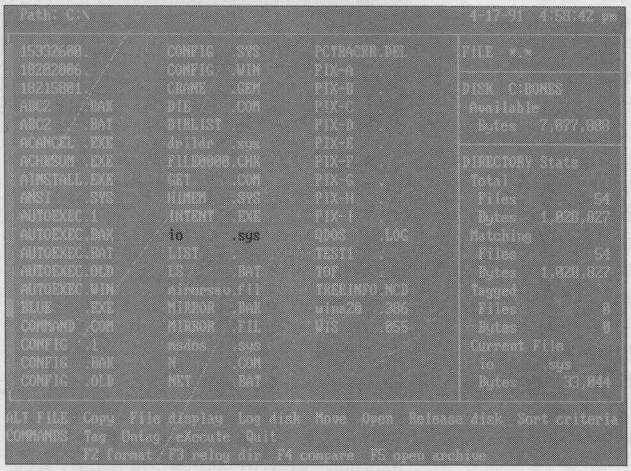
If you have a floppy disk-based PC, the making of system disks is an essential process if you don't want the whole business of computing to be unnecessarily difficult to manage. Hard disk users might find themselves using the facility less frequently, but they will find that occasionally the need does arise and it most definitely helps to know just how to cope with such a situation (see *Panic Disk* on page 248).

BOOTSTRAPPING

A 'system' disk is any disk which can be used to 'bootstrap' (or start up) your PC in the same way as your MS-DOS disk can. There are any number of reasons why you might want to put together such a disk, but the main one is specifically for the floppy disk-based system. If you start up your PC with the MS-DOS disk which was shipped with the computer, you will find that you have to remove that disk and replace it with a program disk containing your application. Then, from time to time, your computer will demand that you replace the MS-DOS disk for a few moments while it performs a particular task (most commonly printing).

All this floppy disk juggling is bad for the disks, which are vulnerable to damage, and the more you shift them around, the more likely they are to corrupt their content. Of course, it can also be a pretty irritating process for you, the user, when all you want to do is concentrate on getting on with the particular job in hand. By creating a system disk and then adding your program files, you can dispense with all that unnecessary and time-consuming fiddling about and concentrate on using your computer for its intended purpose.

When you 'reboot' your PC with the



```
Path: C:\                                                   4-17-91  4:58:42 pm
 15332600.          CONFIG   .SYS     PCTRACKR.DEL     FILE  *.*
 18202806.          CONFIG   .WIN     PIX-A    .
 18215801.          CRAME    .GEM     PIX-B    .       DISK  C:BONES
 ABC2     .BAK      DIE      .COM     PIX-C    .        Available
 ABC2     .BAT      DIRLIST  .        PIX-D    .         Bytes   7,077,888
 ACANCEL  .EXE      drildr   .sys     PIX-E    .
 ACHKSUM  .EXE      FILE0000.CHK      PIX-F    .       DIRECTORY Stats
 AINSTALL.EXE       GET      .COM     PIX-G    .        Total
 ANSI     .SYS      HIMEM    .SYS     PIX-H    .         Files            54
 AUTOEXEC.1         INTENT   .EXE     PIX-I    .         Bytes    1,028,827
 AUTOEXEC.BAK       io       .sys     QDOS     .LOG     Matching
 AUTOEXEC.BAT       LIST     .        TEST1    .         Files            54
 AUTOEXEC.OLD       LS       .BAT     TOF      .         Bytes    1,028,827
 AUTOEXEC.WIN       mirorsav.fil      TREEINFO.NCD      Tagged
 BLUE     .EXE      MIRROR   .BAK     wina20   .386      Files             0
 COMMAND  .COM      MIRROR   .FIL     WIS      .055      Bytes             0
 CONFIG   .1        msdos    .sys                       Current File
 CONFIG   .BAK      N        .COM                         io       .sys
 CONFIG   .OLD      NET      .BAT                         Bytes       33,044
ALT FILE  Copy  File display  Log disk  Move  Open  Release disk  Sort criteria
COMMANDS  Tag  Untag  eXecute  Quit
          F2 format  F3 relog dir  F4 compare  F5 open archive
```

● Although you can't list hidden system files using standard MS-DOS commands such as DIR, a special file management package such as *XTree* will find them

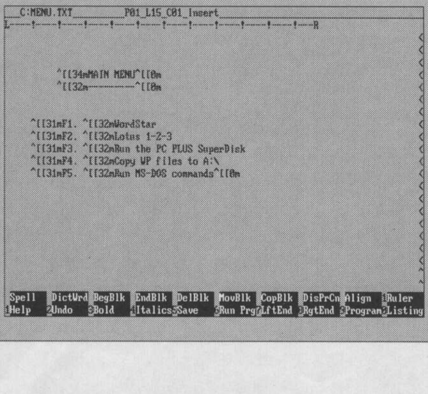
PROJECT MENU – PART THREE

Last month we added some attractive colours to our menu and made it accept the input from Function Keys for the menu options, all with the aid of ANSI.SYS. The menu 'program' (let's bite the bullet – what we're doing here is actually a primitive piece of programming, like it or not) looks acceptable to the user, but we can improve its appearance still further.

The main problem is our menu – despite its many smart colours and custom prompts, it is squashed into the top left hand corner of the screen, and would look considerably better and a lot more a professionally-produced if it occupied the full screen and was centred on the display.

To arrange that, firstly fire up your word processor or text editor in such a way that you can work with plain ASCII files (just like with last month's examples, EDLIN isn't really good enough for the job since you're going to be doing a certain amount of messing around with a number of ESC codes). Using the ECHOed lines of your MENU.BAT file as a rough template, 'paint' or type your menu on to the word processor's screen using the ANSI characters just as before. Be certain to remember that the PC's display is 80 characters wide and 25 lines deep – in fact, it would be very wise not to make your menu screen any deeper than 22 characters for certain reasons which will become clear in a minute.

Your final screen should look something the one below (although



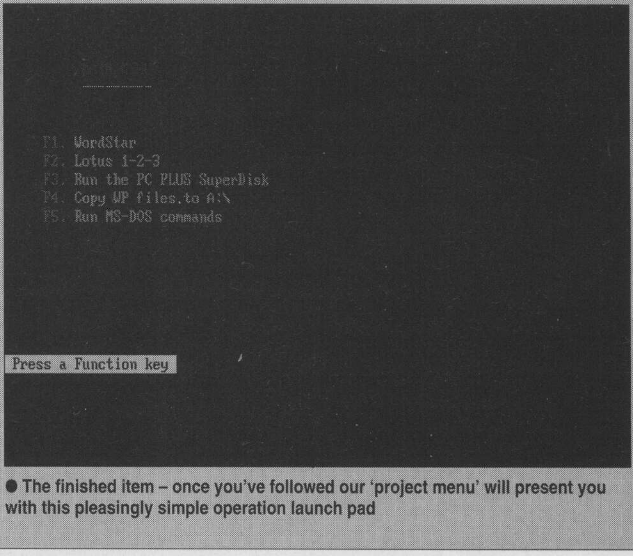
```
   C:MENU.TXT          P01 L15 C01 Insert
L-----!-----!-----!-----!-----!-----!-----!-----!-----!-----!-----!---R



          ^[[34mMAIN MENU^[[0m
          ^[[32m---------^[[0m


     ^[[31mF1. ^[[32mWordStar
     ^[[31mF2. ^[[32mLotus 1-2-3
     ^[[31mF3. ^[[32mRun the PC PLUS SuperDisk
     ^[[31mF4. ^[[32mCopy WP files to A:\
     ^[[31mF5. ^[[32mRun MS-DOS commands^[[0m
```

● These are the lines of code that tell our menu program exactly what entries we want to appear on our list of options. Our editing was carried out in WordStar's non-document mode, but most editors will let you achieve the same results.



```
    F1. WordStar
    F2. Lotus 1-2-3
    F3. Run the PC PLUS SuperDisk
    F4. Copy WP files.to A:\
    F5. Run MS-DOS commands







Press a Function key
```

● The finished item – once you've followed our 'project menu' will present you with this pleasingly simple operation launch pad

MS-DOS disk in the A-drive, the machine will first load two files called IO.SYS and MSDOS.SYS (they are actually called IBMBIO.COM and IBMDOS.COM if you have True Blue IBM PC-DOS, but they come to the same thing).

Next, the file COMMAND.COM is loaded into your PC's memory (RAM); this is the PC's 'command interpreter' and on any occasion that you enter a command at the DOS-prompt, it intercepts it and directs it to the appropriate place – this will be itself. In the case of the commonly-used 'internal' MS-DOS commands such as DIR and COPY, or to an external MS-DOS program for commands such as FORMAT and XCOPY, or to another program entirely such as a word processor or spreadsheet.

If you perform a DIR on your MS-DOS system disk, you will find COMMAND.COM, but not IO.SYS or MSDOS.SYS (or their Big Blue counterparts); this is because they are 'hidden' files – so ensure that you do not forget that they are there. To copy them on to a new disk you cannot use the COPY command because they have to reside in a special area at the very beginning of the disk known as the 'boot sector'. Therefore, you will have to use the SYS command.

COPYING THE SYSTEM

Using the MS-DOS SYS command is simplicity itself, all you have to do is ensure that you are in your MS-DOS directory and then, assuming that you want to copy the system files to the A-drive, enter:

```
SYS A:
```

What happens next rather depends on what release of MS-DOS you have – and the hardware manufacturer's alterations to that release. At the very least, you will have IO.SYS and MSDOS.SYS copied over to your target disk, followed by the message 'system copied'. In some cases, COMMAND.COM will be copied as well, but most systems expect you to do that yourself. The process itself is easy enough – just use the MS-DOS COPY command in the normal way.

Having done that, you can copy or create an AUTOEXEC.BAT and CONFIG.SYS file (as described in earlier episodes of *Beginners' Workshop)*, and Bob's your uncle. In most cases, you will now have a system disk with plenty of room to take your program files.

If you are starting from scratch with a brand new disk, you don't have to use the SYS command. Instead, when you are formatting the disk – using the MS-DOS FORMAT command to prepare the disk for MS-DOS – you can use the /S switch. So the following will format the disk in the A-drive and copy the system files over to it:

```
FORMAT A: /S
```

●

THE PANIC DISK

For a hard disk-owning PC user, the SYS command comes into its own when building a 'panic disk'. This is probably the most important floppy disk you own, and if you don't have one, start making one as soon as you finish reading this article.

The panic disk contains a mirror image of a known, safe configuration as seen on your hard disk – the system files, the CONFIG.SYS, the AUTOEXEC.BAT and any other subsidiary files which they use, such as ANSI.SYS. When (not 'if') you make a mistake and destroy the contents of your hard disk, you can use the panic disk to reboot the PC from floppy, and you can be certain that you have got a solid environment to use as the basis for rebuilding your hard disk.

CHANGING CONFIGURATION

Both hard and floppy disk users may find that a collection of system disks can be very useful if they are constantly having to boot up with different CONFIG.SYS files and AUTOEXEC.BATs – with and without a RAM disk perhaps, or with and without the special files that *Windows* requires.

A special boot disk made up for each different configuration can be slotted in the floppy disk drive, a swift [Ctrl][Alt][Del] and you've changed it all around – this makes things a great deal tidier than having to edit your CONFIG.SYS and AUTOEXEC.BAT files every time you want a change.

it will have your own menu choices and you may well have chosen to customise the menu colours in a different way):

```
^[[34mMAIN MENU^[[0m
^[[32m———-^[[0m
^[[31mF1. ^[[32mWordStar
^[[31mF2. ^[[32mLotus 1-2-3
^[[31mF3. ^[[32mRun the PC PLUS SuperDisk
^[[31mF4. ^[[32mCopy WP files to A:\
^[[31mF5. ^[[32mRun MS-DOS commands^[[0m
```

It will look best if it is at least 15 lines long. Save the file as MENU.TXT and make sure it is in your \MENU subdirectory (copying it there if that is necessary).

Now load MENU.BAT into a word processor or text editor and alter the opening section so it looks like this:

```
@ECHO OFF
C:
CD\MENU
CLS
TYPE MENU.TXT
PROMPT
```

(Type the following two lines as a single line)

```
[0;59;"1";13p$e[0;60;"2";13p$e[0;61;"3";13p$e[0;62;
"4";13p$e[0;63;"5";13p$e[47;34m Press a Function key $e[0m
```

If you follow through the logic of this it should be clear what we have done: by removing the displayed menu from the batch file, we have removed the restrictions imposed by that environment. The batch file now TYPEs our elegant centred menu on to the screen and then places the prompt beneath it.

Functionally nothing has changed – it is exactly the same as it was before, but now it is a good deal more aesthetically pleasing. At this point it should become clear why the menu has to have fewer than 25 lines – if there were any more, the top of it would end up scrolling off the screen.

By now, MENU.BAT should provide you with a reasonable imitation of a 'proper' menuing program – it is time for you to use it and thoroughly enjoy it! If you like messing around with batch files at this level you could do worse than to buy Van Wolverton's *Supercharging MS-DOS* book, which is currently available from the Microsoft Press. Among other very useful utilities, he tells you how to make a menu program which is even better than this one.

HELP US TO HELP YOU

hy not take this opportunity to ask l those niggling questions to which ou have been unable to find a atisfactory answer.

There's space here for three but if our list is longer, please write them n a separate sheet. Answers will ther be incorporated into future rticles or answered on this page.

....

....

....

....

....

....

Please clip out this form (or use a photocopy) and send it to:

Beginners' Workshop,
PC PLUS,
Beauford Court,
30 Monmouth Street,
Bath
BA1 2BW.

Please print clearly:

Name..

Daytime Tel......................................